

# 5.服务端接口开发

---

1.对Hr类进行Security接口开发

2.创建HrService实现简单登录

## 1.对Hr类进行Security接口开发

实现 `UserDetails` 接口以及方法

```

public class Hr implements UserDetails {

    /** hrID */

    private Integer id;
    /** 姓名 */

    private String name;
    /** 手机号码 */

    private String phone;
    /** 住宅电话 */

    private String telephone;
    /** 联系地址 */

    private String address;
    /** */

    private Boolean enabled;
    /** 用户名 */

    private String username;
    /** 密码 */

    private String password;
    /** */

    private String userface;
    /** */

    private String remark;

```

```

    public Integer getId() { return id; }

    public void setId(Integer id) { this.id=id; }

    public String getName() { return name; }

    public void setName(String name) { this.name=name; }

    public String getPhone() { return phone; }

    public void setPhone(String phone) { this.phone=phone; }

    public String getTelephone() { return telephone; }

    public void setTelephone(String telephone) { this.telephone=telephone; }

    public String getAddress() { return address; }

    public void setAddress(String address) { this.address=address; }

    public void setEnabled(Boolean enabled) { this.enabled=enabled; }

    public String getUsername() { return username; }

    public String getPassword() { return password; }

    public void setPassword(String password) { this.password=password; }

    public String getUserface(){...}public void setUserface(String userface){this.userface=userface;}

```

```
public String getRemark(){return remark;}

public void setRemark(String remark) { this.remark=remark; }
```

```
@Override
public boolean isAccountNonExpired() { return true; //账户是否没有过期 }

@Override
public boolean isAccountNonLocked() { return true; //账户是否没有被锁定 }

@Override
public boolean isCredentialsNonExpired() {
    return true; //密码是否没有过期
}

@Override
public boolean isEnabled() { return enabled; }

public void setUsername(String username) { this.username=username; }

@Override
public Collection<? extends GrantedAuthority> getAuthorities() { return null; }
}
```

## 2.创建HrService实现简单登录

```

2 个用法
@Service
public class HrService implements UserDetailsService {
    1 个用法
    @Autowired
    HrMapper hrMapper;

    @Override
    public UserDetails loadUserByUsername(String username) throws UsernameNotFoundException {
        Hr hr=hrMapper.loadUserByUsername(username);
        if (hr==null){
            throw new UsernameNotFoundException("用户不存在");
        }
        return hr;
    }
}

```

```

3 个用法
@Mapper
public interface HrMapper{

    1 个用法
    Hr loadUserByUsername(String username);
}

```

```

//resultmap
<select id="getAllHrRoles" resultMap="BaseResultMap">
    select * from hr_role;
</select>

```

创建SecurityConfig对Security简单配置

```

Java | 复制代码

1  @Configuration
2  public class SecurityConfig extends WebSecurityConfigurerAdapter {
3      @Autowired
4      HrService hrService;
5
6      @Bean
7      PasswordEncoder passwordEncoder(){
8          return new BCryptPasswordEncoder();
9      }
10
11     @Override
12     protected void configure(AuthenticationManagerBuilder auth) throws
Exception {
13         auth.userDetailsService(hrService);
14     }
15 }

```

创建一个消息提示类(get set方法省略)

```

public class RespBean {
    3 个用法
    private Integer status;
    3 个用法
    private String msg;
    3 个用法
    private Object obj;

    1 个用法
    public static RespBean ok(String msg){
        return new RespBean( status: 200,msg, obj: null);
    }

    1 个用法
    public static RespBean ok(String msg,Object obj) { return new RespBean( status: 200,msg,obj); }

    2 个用法
    public static RespBean error(String msg) { return new RespBean( status: 500,msg, obj: null); }
    public static RespBean error(String msg,Object obj) { return new RespBean( status: 500,msg,obj); }

    public RespBean() {
    }

    4 个用法
    public RespBean(Integer status, String msg, Object obj) {
        this.status = status;
        this.msg = msg;
        this.obj = obj;
    }
}

```